(2)

AD A124848

COMPUTER SCIENCE
TECHNICAL REPORT SERIES

DTIC
ELECTE
FEB 24 1983
S D
B

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DTIC FILE COPY

83 02 023 132

TR-1185
DAAG-53-76C-0138

July, 1982

USING PYRAMIDS TO DETECT GOOD CONTINUATION

T. H. Hong
M. Shneier
R. Hartley
A. Rosenfeld

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

ABSTRACT

Pictures containing lines and curves are often perceived by humans as being composed of a smaller number of more global figures. For example, a broken set of collinear line segments is perceived as a single straight line, even in the presence of other overlapping line segments. This paper presents a method of extracting such figures from images automatically in a highly parallel manner. A pyramid of successively lower-resolution images is used to transform the problem from one of global search to one of local good continuation. Using the pyramid, figures that are both sparse and obscured can successfully be extracted and the background clutter can be suppressed.

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (DARPA Order 3206) is gratefully acknowledged, as is the help of Janet Salzman in preparing this paper.

1. Introduction

When humans analyze pictures, they apply a number of grouping rules to the data. These rules were discussed by Gestalt psychologists, and include proximity, closure, good continuation, and similarity grouping (Koffka, 1935). This paper addresses the application of pyramid processes to good continuation grouping for the cases of sparse and broken collinear line segments and smooth curves. Earlier papers have examined the use of pyramids for grouping based on closure (Hong et al., 1981; Hong and Shneier, 1981).

Important properties of the pyramid representation are that it brings features that were originally far apart into close contact, and, at the same time, it reduces the differences between features. Both the current and the earlier work take advantage of these properties to group features based on proximity and similarity at a number of resolutions. By searching in local neighborhoods for particular patterns and ignoring points that do not match, it becomes possible to extract the desired features even if the evidence for their existence is sparse and they are obscured by other features.

The problem to be addressed in this paper involves extracting the straight lines and smooth curves in an image. The image may contain both straight and curved segments, which may overlap. The straight-line segments may be broken and sparsely distributed, but the curves are assumed to be largely complete, with only small gaps. (It is not always obvious how to link up curves that have

large gaps). The method to be presented uses local position, direction, and curvature information at a number of resolutions to construct a linked pyramid.

The pyramid is based on 4x4 overlapping neighborhoods, so that each node has 4 "fathers" on the level above it, and 16 "sons" on the level below it, when these levels exist. Each father contains a summary of the information in its central 2x2 block of sons. For each curve that passes through this block, the two endpoints are stored (at full resolution). The directions at the endpoints are stored also. A father can store a fixed maximum number of curves, after which a flag is set to denote an overload. As the resolution decreases, there are fewer and fewer nodes available to describe all the curves, and more and more nodes contain intersecting curves. Rules have been developed for discarding curves in such a way that only the straight lines and smooth curves survive.

The next section describes the algorithm in detail. It is followed by a series of examples of the output for various images and a discussion of the techniques involved.

2. Extracting lines and curves

Straight lines and curves are perceived by humans even if they are obscured by other curves, or are broken and relatively sparse. The amount of obscuration and missing information allowed depends on the degree of curvature of the lines. Conventional image processing techniques could be devised to extract these figures from images, by searching for pairs of curve segments that continue one another, but at great computational cost due to the global nature of the problem.

The advantage of the pyramid-based technique is that global properties can be made local, if the pyramid is constructed in the right way. The difficulty is that reducing the resolution of the image, if done naively, can cause essential information to be lost. By *summarizing* the information in an informed way based on the desired results, an efficient technique can be developed for extracting the curves. Different techniques can be devised for different curves. Those to be described here can be adapted to extract straight lines, smooth curves, or arcs of circles.

The input is an image that contains straight and curved lines which may overlap. A line can have significant gaps, and it will still be extracted as a unit, so long as it is straight. Curves cannot have large gaps, but can be crossed by other curves. Figure 1 shows examples of images that were processed successfully.

The algorithm works in a single pass, starting at the lowest level of the pyramid (the original image). Each successive level is constructed on the output of the previous level, until the top level is reached. The information in the top level(s) is then used to display the curves that were extracted. The levels in the pyramid are numbered so that the original level is level 0, the next level is level 1, and so on. A node at level L has sixteen sons at level L-1, arranged in a 4x4 block. Adjacent blocks overlap by two rows and two columns in each direction, so that each son has four fathers (Figure 2). The central 2x2 region of each 4x4 block is unique to that block, although each element of the 2x2 region contributes to three other regions as well.

Each node in the pyramid contains three pieces of information. First is a counter or flag that insures that the capacity of the node to store lines is not exceeded. In the implementation, the maximum was set at 5. Second, the coordinates of the endpoints of the curves are stored (at full resolution). If a curve extends outside the central 2x2 region of a neighborhood, its extremal points within the borders of the 2x2 region are stored as its endpoints. Third, a measure of the direction at each endpoint is stored.

Level 0 of the pyramid consists of the original image. Each node at each level i (i=1,...,L) has a total of 16 sons, but only 4 central sons, labelled c_1, c_2, c_3, and c_4 in Figure 3.

It is only these sons that are eligible to have their information passed up to their father. Because the central 2x2 regions cover the image all points are eligible to be passed up to some father. A further requirement is that a curve be supported by some other curve in its neighborhood. Support is measured using a merit function to be defined below.

Any of the 16 sons in a neighborhood can support the passing up of information from one of the central sons, but a curve cannot give support to more than one other curve in the central region.

For each curve in each of the central 2x2 nodes, a local search is applied to find support for passing the curve up to its father. The figure of merit is calculated between pairs of facing endpoints, one from each curve. It requires that the curves merge smoothly and are not too far apart. The figure of merit will be discussed in detail below.

When a curve in the central 2x2 region finds support, there are two cases to be considered when deciding how to pass it up to the father node. The first arises when the support originates outside the central 2x2 region, and the second when it comes from within the region. If the support is from outside the region, then the curve is not extended but is passed up intact to its father (i.e., the endpoints are not changed, and neither are their direction measures). The curve that provided support is not included in the information that is passed up. This does not cause a loss of information because the supporting curve is a member of some other 2x2 central region, and will be passed to its

father because the support relation is symmetric.

When support arises from within the central 2x2 region, however, it becomes necessary to merge the supporting curves. This is done in the simplest possible way. The endpoints of the curves are examined, and the two that are furthest apart (at opposite ends of the curve) are chosen to represent the whole curve. These endpoints and their associated direction measures are passed up to their father. Note that it is not necessary to recalculate the positions or directions of the points. It is also unnecessary to know the shape of the curve between the endpoints, although this can be recovered by tracing down through the pyramid.

The merit function used to define the support relation is based on fitting a circle with some given radius c to the region between the endpoints of two adjacent curves. In Figure 4, let d be the length of the straight line joining p_1 and p_1', and let d' be the length of the arc of the circle joining p_1 and p_1'.

$$\text{Then } \hat{\theta} = \frac{d'}{c} \text{ radians}$$

$$= \cos^{-1}\left(1-\frac{d^2}{2c^2}\right)$$

Let ϵ be the total measurement error (digitization error and error in slope measurement). Let θ be the actual difference between the slopes of the curves at p_1 and p_1'. There are three variants of the merit function that can be defined, depending on whether straight lines, smooth curves, or arcs of circles are to be detected.

To detect straight lines, set $c=\infty$ (a large number), so that $\hat{\theta}=0$. Then if $(|\theta| \leq (\hat{\theta}+\epsilon))$ the lines p_1p_2 and $p_1'p_2'$ are collinear, otherwise they are not.

To detect curves with radius of curvature $<1/c$, if $(|\theta| \leq (\hat{\theta}+\epsilon))$ then curves p_1p_2 and $p_1'p_2'$ can be merged into the same curve (p_2p_2'); otherwise they are different curves.

To detect arcs of circles with radius c, the following condition must be satisfied:

$$(|\theta| < (\hat{\theta}+\epsilon/2)) \text{ and } (|\theta| > (\hat{\theta}-\epsilon/2)).$$

In this case, p_1p_2 and $p_1'p_2'$ are part of the same circle; otherwise they are not.

The figure of merit is applied to each pair consisting of a curve from the central 2x2 region and curves in compatible grid positions in the 4x4 neighborhood. Another merit function that was tried (Broder and Rosenfeld, 1981) was not general enough to deal with curves, and produced much poorer results when applied to straight lines.

If a node is asked to represent more than the maximum number of curves, it sets a flag denoting overload, and stops accepting curves. These latter curves could in principle be recovered by examining the sons of a node, but in practice that has not been found necessary. The maximum number of curves that can be accepted was set at 5 in the implementation (see Miller, 1956). The order in which curves are accepted is the order in which they are passed up. No filtering is applied to the lines

based on smoothness or length, although this might result in improved performance.

The process is repeated at each level of the pyramid, until the top level is reached. At this stage, the remaining lines and curves can be displayed. If only the top level is displayed, a maximum of 5 lines can be found. By going down to lower levels, more curves can be displayed. Curves can be displayed by simply drawing straight lines between their endpoints, giving a rough estimate of their position and direction. They can also be displayed at full resolution by tracing links through the pyramid to the bottom level. The next section shows examples of some images that were processed.

3. Examples

Figure 1 shows images of curves and lines to be processed. Figure 5 illustrates the processing for extracting straight lines. The original image consists of three crossing, broken, straight lines. For this example, the radius c in the merit function was set to a large number (1000) to favor collinear segments. The bottom left picture is the original image, and the other three pictures show the contents of levels 3,4, and 5 of the pyramid. It is clear that small errors at points of intersection are very soon corrected, and the final result (top right) is what is desired. Level 6, the top level, is not shown. At this level there are three surviving lines, represented by their endpoints, and corresponding to the three lines perceived by humans.

Figures 6 and 7 show how variants of the merit function affect the results of processing curves. Figure 6 shows a sinusoidal curve intersected by a straight line. The curve is not perfect, and the line is broken. The processing in Figure 6 reflects the program's attempts to segment the image into straight line segments, using a value of 1000 for c, as above. This gives a good approximation to the curve, up to level 5, although level 6 would show a straight line between the endpoints.

In Figure 7, the same curve as in Figure 6 is processed, but this time with a value of 4 for c, allowing a large amount of curvature. The result is that the curve merges more rapidly

into a single segment (represented by the lines joining its endpoints). Note that although the display appears to show that the shape of the curve has been lost, this is not actually the case. By projecting nodes down through the pyramid, it would be possible to display the curve at its full resolution. What is being displayed is the process of merging segments of the curve into a single component.

Figures 8 and 9 illustrate a fundamental ambiguity in interpreting broken or crossing curves. In many cases, there is more than one interpretation that explains the data. For example, in Figure 8 the curves are interpreted as intersecting sinusoids and are segmented accordingly into two relatively straight pieces. In Figure 9, however, the same data is interpreted as two curves that, approach each other but do not cross, leading to a totally different segmentation. By altering the criteria on the curvature in the merit function, one or the other of these interpretations will be favored.

Figure 10 shows an example of the processing of a circle of known radius (c=10). The successive levels clearly show how the curve is summarized. The top level (not shown) represents the circle as a single component, which can be linked through the levels of the pyramid to the whole circular boundary.

Shneier (1980) describes a method of extracting linear features from images using pyramids. A line detector was run at each level of the pyramid, and the lines detected were used as plans to extract the linear features by thresholding. The procedure extracted lines from the gray-scale pyramid, and did not use linking of any kind between levels.

In an approach similar to that used here, Shneier (1981) describes two hierarchical representations for linear features or curves. In these representations, a 4x4 overlapped region was used as the basis of constructing the description, but only one line was allowed to be represented at the next highest level for each such neighborhood. In addition to the magnitude and direction of the line, an intercept in the neighborhood and an error in direction were stored. The error was large at corners and intersections, and small for straight segments. This allowed a class of edge quadtrees to be defined that gave a piecewise linear representation of the curves in a linear image, and a class of edge pyramids that represented the information less completely, but retained the important edges at high levels in the pyramid, and discarded the noise. Both representations were able to represent edges or lines.

The method described in this paper could be extended to handle thicker objects in an interesting way if a technique such as that described in Hong and Shneier (1981) were used in parallel

4. Discussion

Pyramids are naturally suited to dealing with compact regions. It is therefore interesting that they can be adapted to extract and describe linear and curved features. Of even greater interest is the ability to distinguish between straight and curved segments, and to select for particular properties. Here this is achieved by adjusting the radius in the merit function.

Earlier work has been concerned with pyramid representations for curves, and with clustering of collinear line segments. Various merit functions have been defined for collinear line segments, such as that described by Broder and Rosenfeld (1981). Scher *et al*. (1982) describe a number of methods for grouping sets of collinear line segments. There the problem is in one dimension, and consists essentially of finding proximity grouping strategies. They describe merging and splitting methods for finding which subsets of line segments can be considered to belong to the same cluster. The problem here is more general because of the two-dimensionality, the possibility of crossing lines or curved lines, and the digitization of the image. The assumption made here about individual straight lines is also somewhat different. Any segments that are collinear will be grouped, no matter how they cluster. If this is undesirable, the methods described in Scher *et al*. can be applied to the individual lines discovered by this process to split further.

with the present technique. Hong and Shneier describe a pyramid-based method of extracting objects from images which reduces the resolution of the objects to find those that are compact or elongated, and then expands those components. Elongated objects are reduced to curves in this method, and the curves could be fed into the appropriate level of the pyramid for the current method. This would then complete the processing as if the object were a line. On projecting back down, it would be necessary to transfer back to the gray-level pyramid at the same level to extract the thick objects.

There are a maximum 5 curves per node, so there could potentially be a large number of compatibility tests to be made. In practice, however, images are not very dense, except near intersections, and even here the maximum is seldom approached. The search can be further reduced because only a small number of search directions could possibly yield compatible curves.

The algorithm presented here is extremely efficient. It makes a single pass, constructing successive pyramid levels, and then displays the lines that it discovers. All processing is local, and takes place in parallel at each level of the pyramid.

5. Conclusions

This paper has presented a method of extracting straight line segments and curves from images in a highly parallel manner using a hierarchical image structure or pyramid. The segments are extracted even if they are broken and obscured by other curves. This kind of performance is similar to that shown by humans in similar situations.

The current work demonstrates the applicability of the pyramid representation to images containing elongated and curved components, in addition to their established use in extracting compact regions. This further demonstrates the power and flexibility of the pyramid approach to image processing.

References

1. T. H. Hong, M. Shneier, and A. Rosenfeld, Border extraction using linked edge pyramids. TR-1080, Computer Science Center, University of Maryland, College Park, MD, July 1981.

2. T. H. Hong and M. Shneier, Extracting compact objects using linked pyramids. TR-1123, Computer Science Center, University of Maryland, College Park, November 1981.

3. A. Broder and A. Rosenfeld, A note on collinearity merit, Pattern Recognition 13, 1981, 237-239.

4. K. Koffka, Principles of Gestalt Psychology, Harcourt, Brace, New York, 1935.

5. G. A. Miller, The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Reviews 63, 1956, 81-97.

6. A. Scher, M. Shneier, and A. Rosenfeld, Clustering of collinear line segments. Pattern Recognition 15, 1982, 85-91.

7. M. Shneier, Extracting linear features from images using pyramids, IEEE Trans. SMC-12, July/August 1982.

8. M. Shneier, Two hierarchical linear feature representations: Edge pyramids and edge quadtrees. Computer Graphics and Image Processing 17, 1981, 211-224.

Figure Captions

Figure 1. Examples of images successfully analyized by the algorithm.

Figure 2. Two neighborhoods used in constructing pyramids. The central 2x2 regions are disjoint, but each neighborhood shares rows and columns with its neighbors.

Figure 3. The sons in a neighborhood eligible to have their contents passed up to their fathers.

Figure 4. Terms used to compute the merit function. p_1p_2 and $p_1'p_2'$ are two curves. If d is the length of the straight line joining p_1 and p_1', d' is the length of the arc joining p_1 and p_1', and c is the radius of the circle, then

$$\hat{\theta} = d'/c = \cos^{-1}(1 - d^2/2c^2)$$

Figure 5. An example of processing broken straight lines (c=1000). a. The original image. b. Level 3 in the pyramid. c. Level 4. d. Level 5.

Figure 6. An example of processing a curve (c=1000) a. The original image. b. Level 3. c. Level 4. d. Level 5.

Figure 7. The same curve as in Figure 6, but with c=4. a. Original image. b. Level 3. c. Level 4. d. Level 5.

Figure 8. Intersecting curves. The merit function was set to favor straight segments. a. Original image. b. Level 3. c. Level 4. d. Level 5.

Figure 9. Intersecting curves. The merit function was set to favor smooth curves. a. Level 1. b. Level 3. c. Level 4. d. Level 5.

Figure 10. Processing a circle (c=10). a. Level 1. b. Level 3. c. Level 4. d. Level 5.

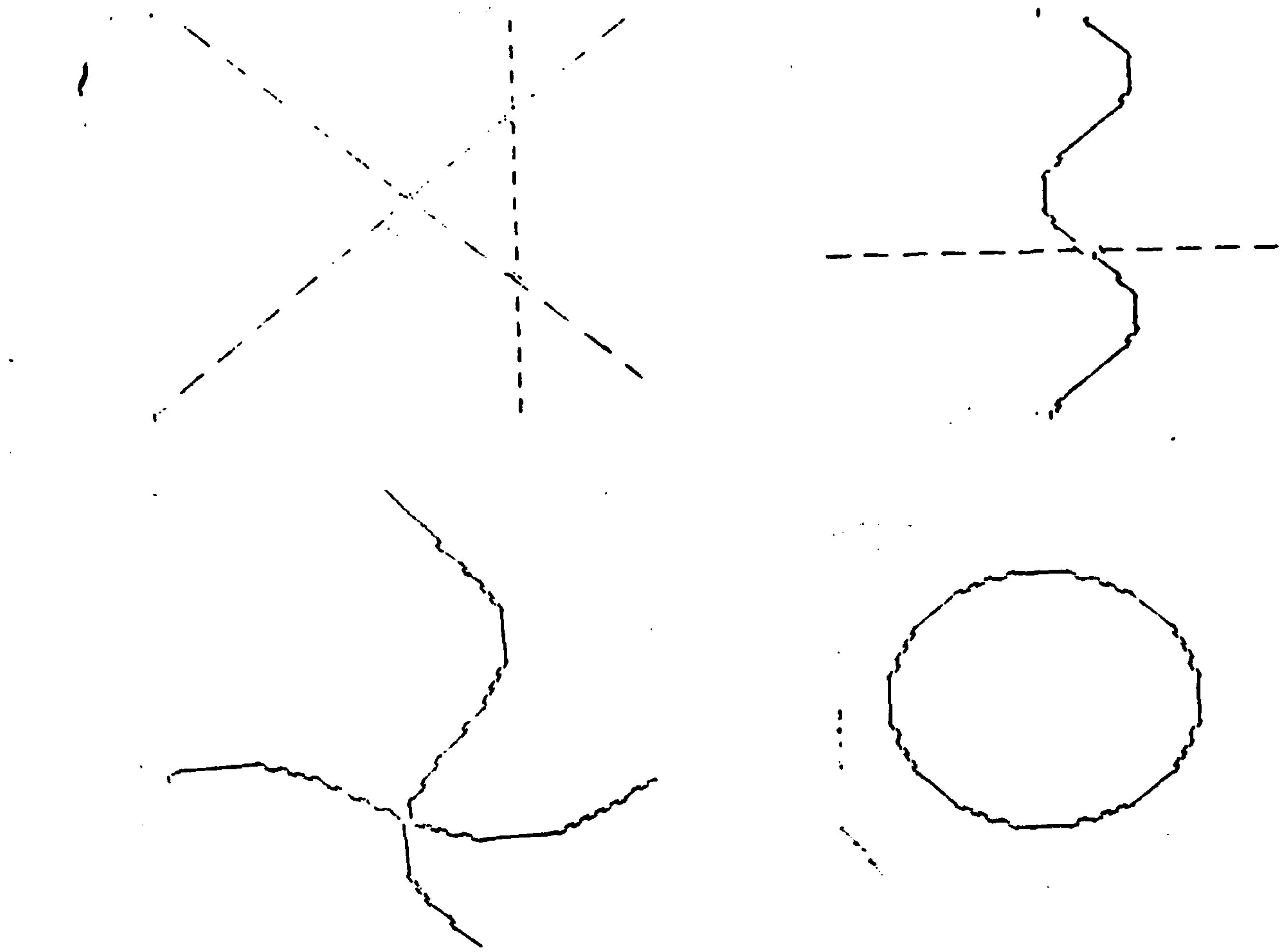


Figure 1

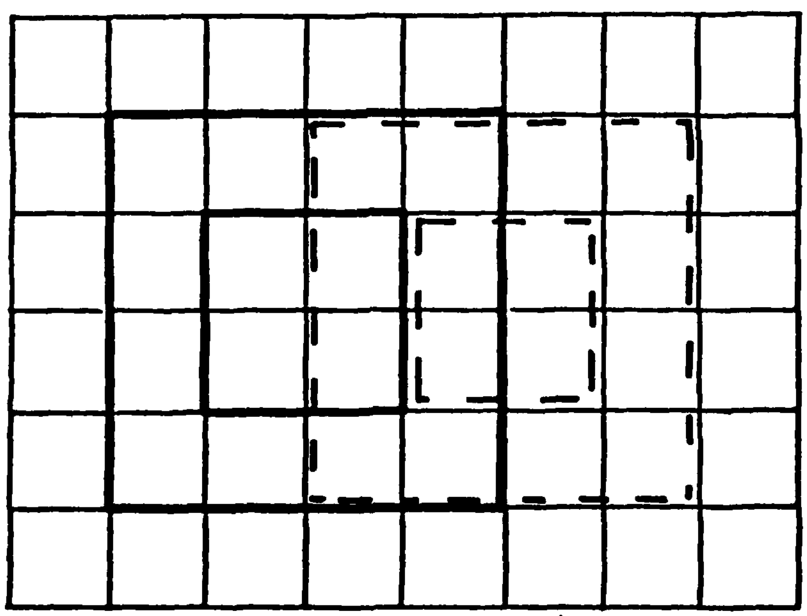


Figure 2

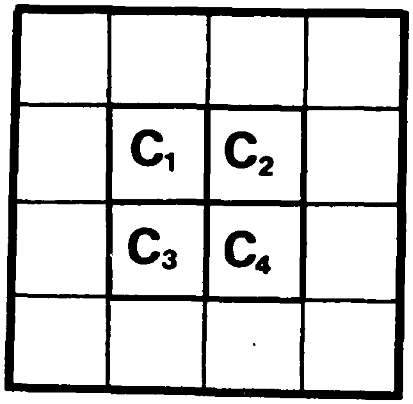


Figure 3

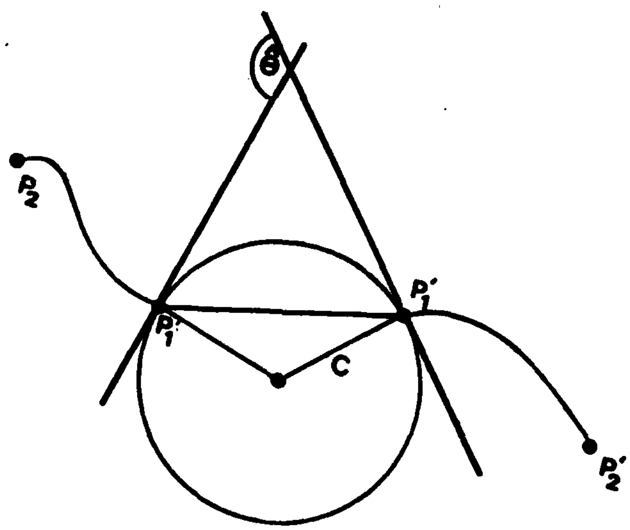
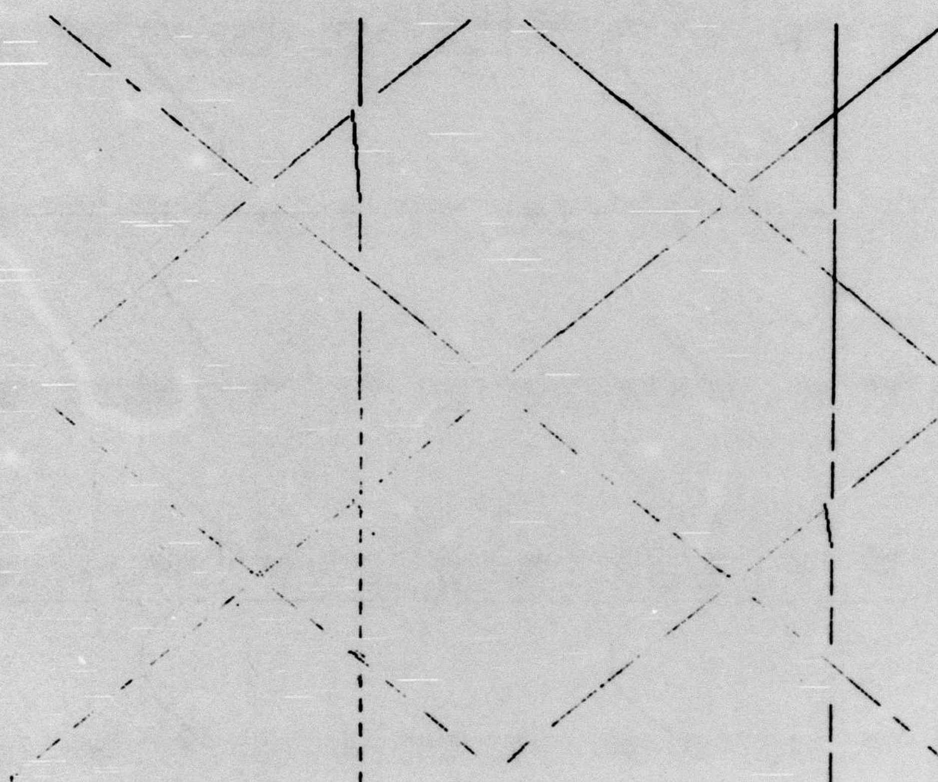


Figure 4



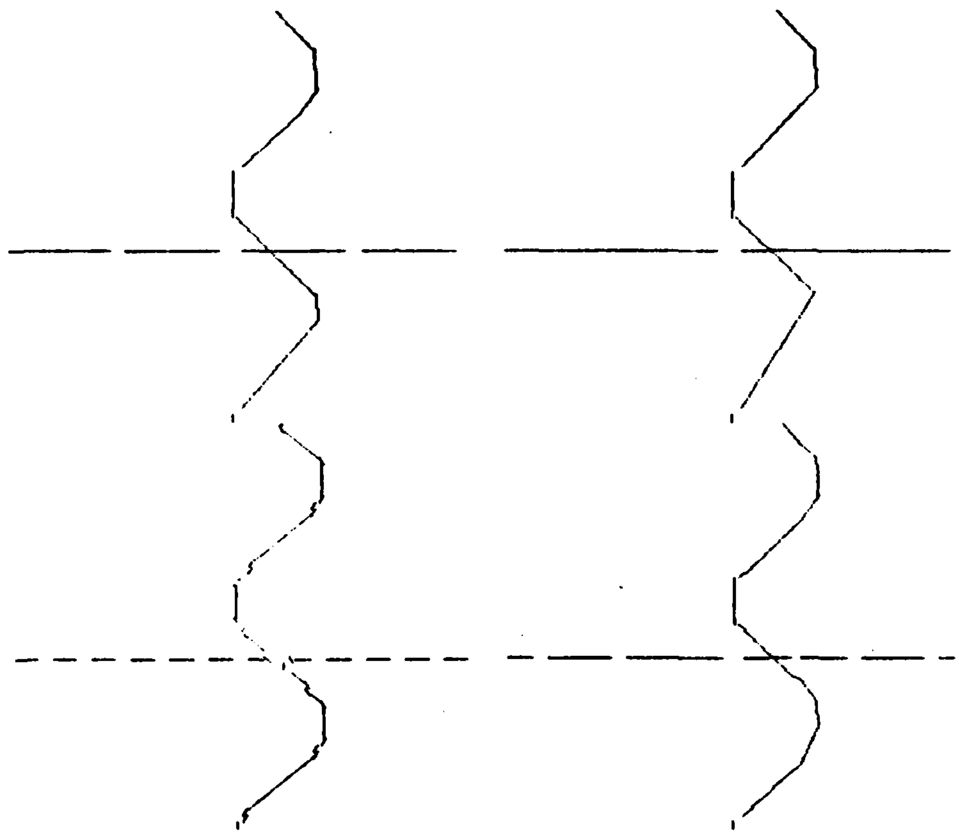


Figure 6

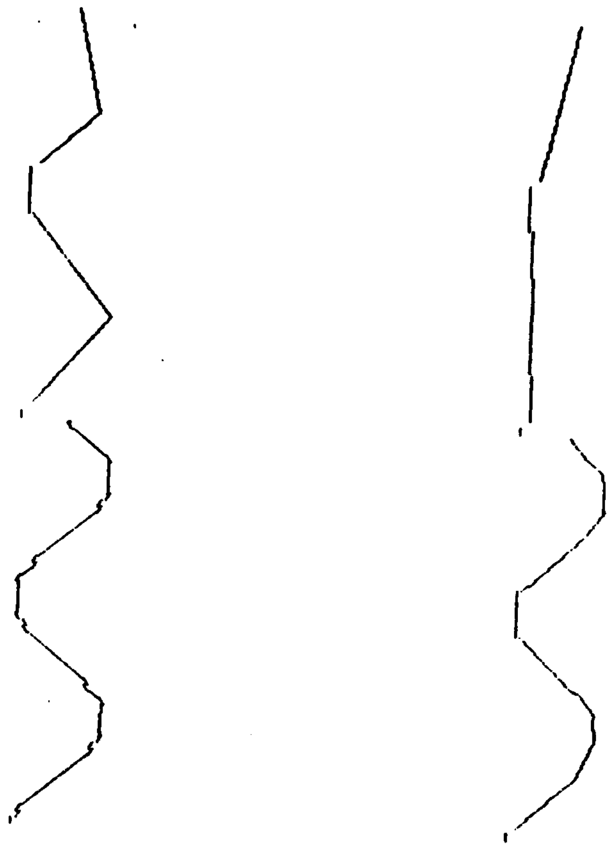


Figure 7

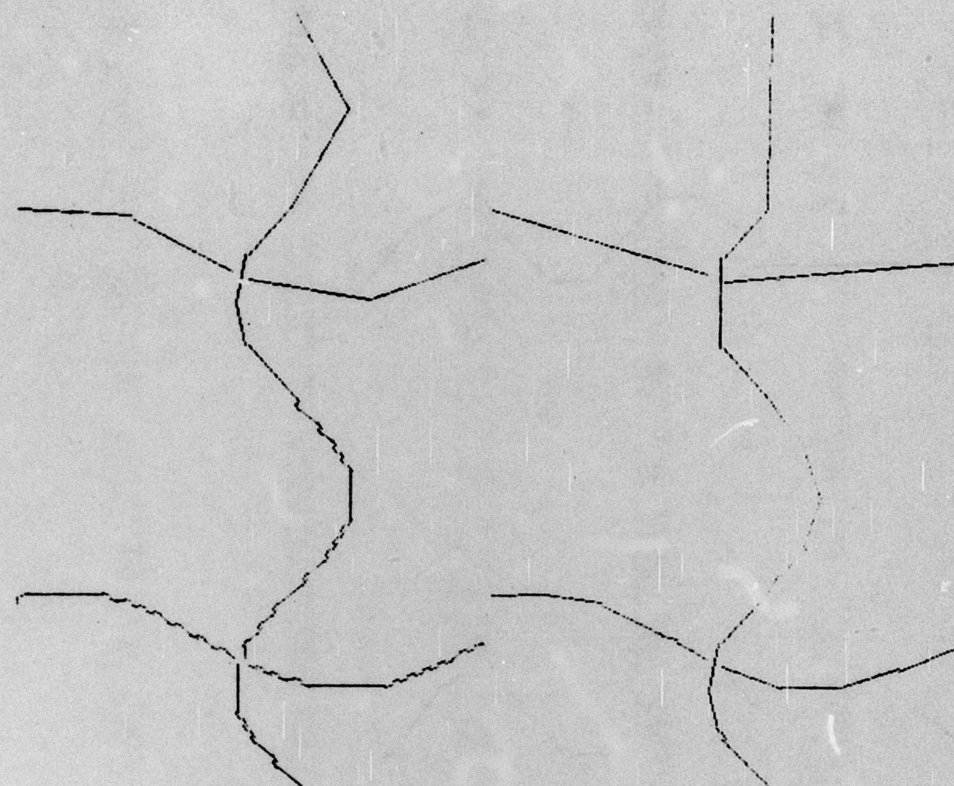


Figure 8

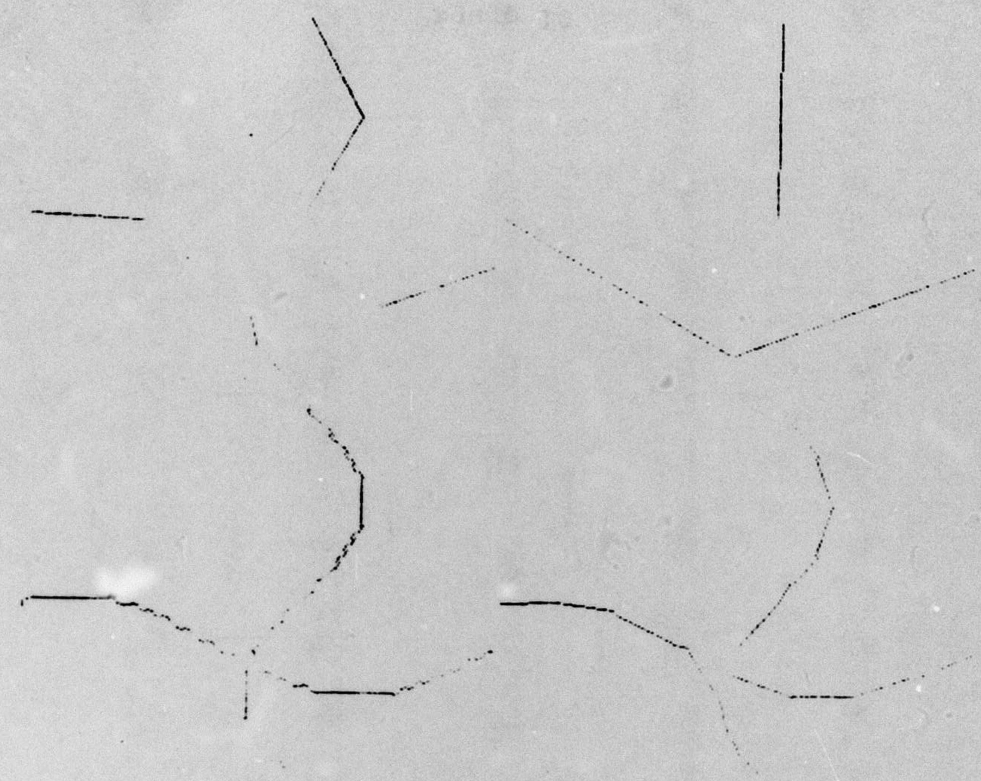


Figure 9

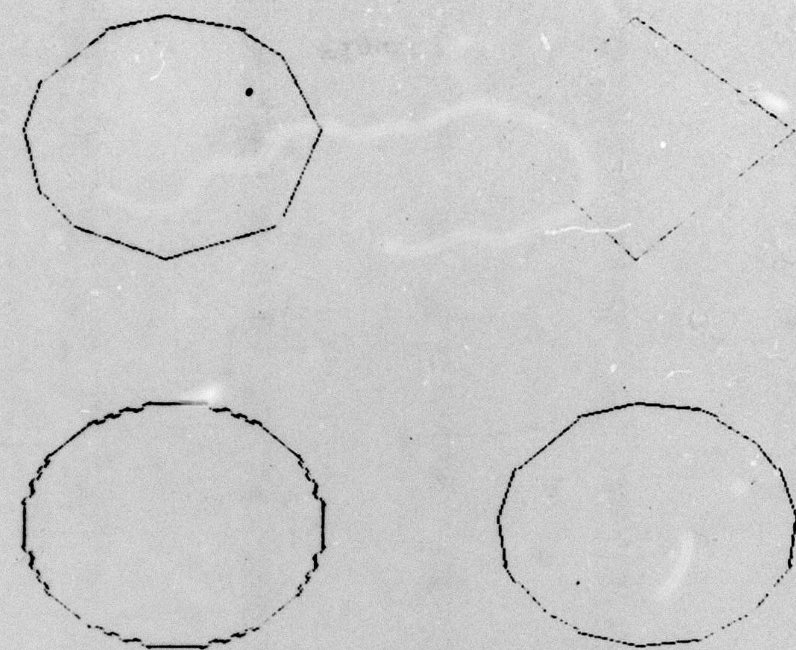


Figure 10

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. A124848
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) USING PYRAMIDS TO DETECT GOOD CONTINUATION	5. TYPE OF REPORT & PERIOD COVERED Technical
	6. PERFORMING ORG. REPORT NUMBER TR-1185
7. AUTHOR(s) T. H. Hong, M. Shneier, R. Hartley, A. Rosenfeld	8. CONTRACT OR GRANT NUMBER(s) DAAG-53-76C-0138
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD 20742	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Night Vision Lab. Ft. Belvoir, VA 22060	12. REPORT DATE July 1982
	13. NUMBER OF PAGES 27
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Image processing
Pattern recognition
Pyramids
Curve detection
Good continuation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Pictures containing lines and curves are often perceived by humans as being composed of a smaller number of more global figures. For example, a broken set of collinear line segments is perceived as a single straight line, even in the presence of other overlapping line segments. This paper presents a method of extracting such figures from images automatically in a highly parallel manner. A pyramid of successively lower-resolution

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

images is used to transform the problem from one of global search to one of local good continuation. Using the pyramid, figures that are both sparse and obscured can successfully be extracted and the background clutter can be suppressed.